Project 1

<Blackjack>

Course: CIS – 5 47948

Name: Moreno, Brenda

Date: 05 November 2018

Introduction:

What is Blackjack? Blackjack, also known as “21,” is a gambling card game.

The objective of the game is to get as close to 21 to defeat the dealer. The player can hit as many times as they’d like; however, too many cards can result in a bust.

Unlike other card games, blackjack uses one to eight decks of cards. Each deck contains 52 playable cards. It is common for multiple players to be seen seated at a blackjack table. Nonetheless, the players do not compete against each other. Each player plays against the dealer, also known as the house. The player must deal to play. From there, the dealer reveals his first card and the player can either hit, split, stand, or surrender.

Once the player has finished their turn, the dealer will turn over his hole card (second card). the dealer’s hand must be greater than or equal to 17, if the dealer stands at a 17 it is called a “soft 17.”

Blackjack Terms:

Blackjack: If the player starts with two cards and equals the value of 21

Bust: Also known as break, the player or dealer have passed 21 and they automatically lose.

Hit: Another card is given to the player.

Stand: Player stands with his cards.

Surrender: Player forfeits. The wager is split in half and the player does not play his/her hand. (Note: *It is common for casinos to not have this option available for the player.)*

Card Value:

Ace: 1 or 11

2-10: The written face value.

Jack, Queen, & King= 10

The Ace card is the only card that carries two available values. If the player draws an Ace, he or she must select the value of the card, either 1 or 11.

Summary:

Project size: 278 lines

The number of variables: approx. 21

The number of methods: 5

In all honesty, Project 1 was a challenge. My end goal was to create a playable blackjack game without the use of advanced concepts such as arrays and functions. In the end, I was able to create something similar but playable.

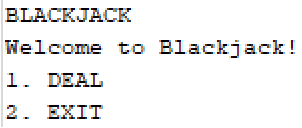
For the simplicity of this version, I only allowed the player to hit or stand. The player will also not be able to select the face value of an Ace card.

I used the seven constructs while constructing the program. While the program was repetitive, it was tedious in where to place specific constructs, for I kept referencing back to the book and the game examples provided in the GitHub.

It took me the two full weeks to complete this project. One of the reasons was because I wanted to create a poker game. The second was to find a method without using advance concepts not yet covered. Third, I was too stubborn to quit to turn it in unplayable. Nonetheless, I am content of my results, and I would like to one day improve my project.

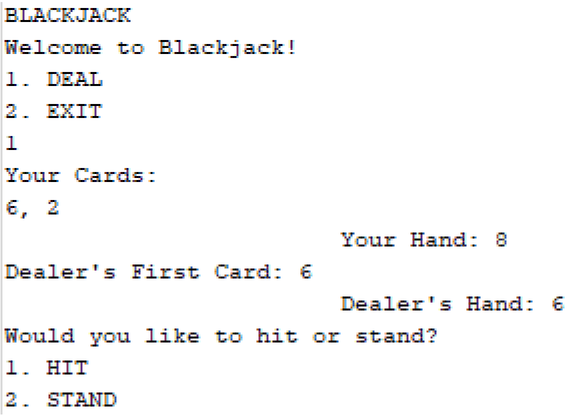
Description:

To play, the player must type in 1 or 2. Deal or Exit. Hit or Stand.



```
BLACKJACK
Welcome to Blackjack!
1. DEAL
2. EXIT
```

After inputting 1

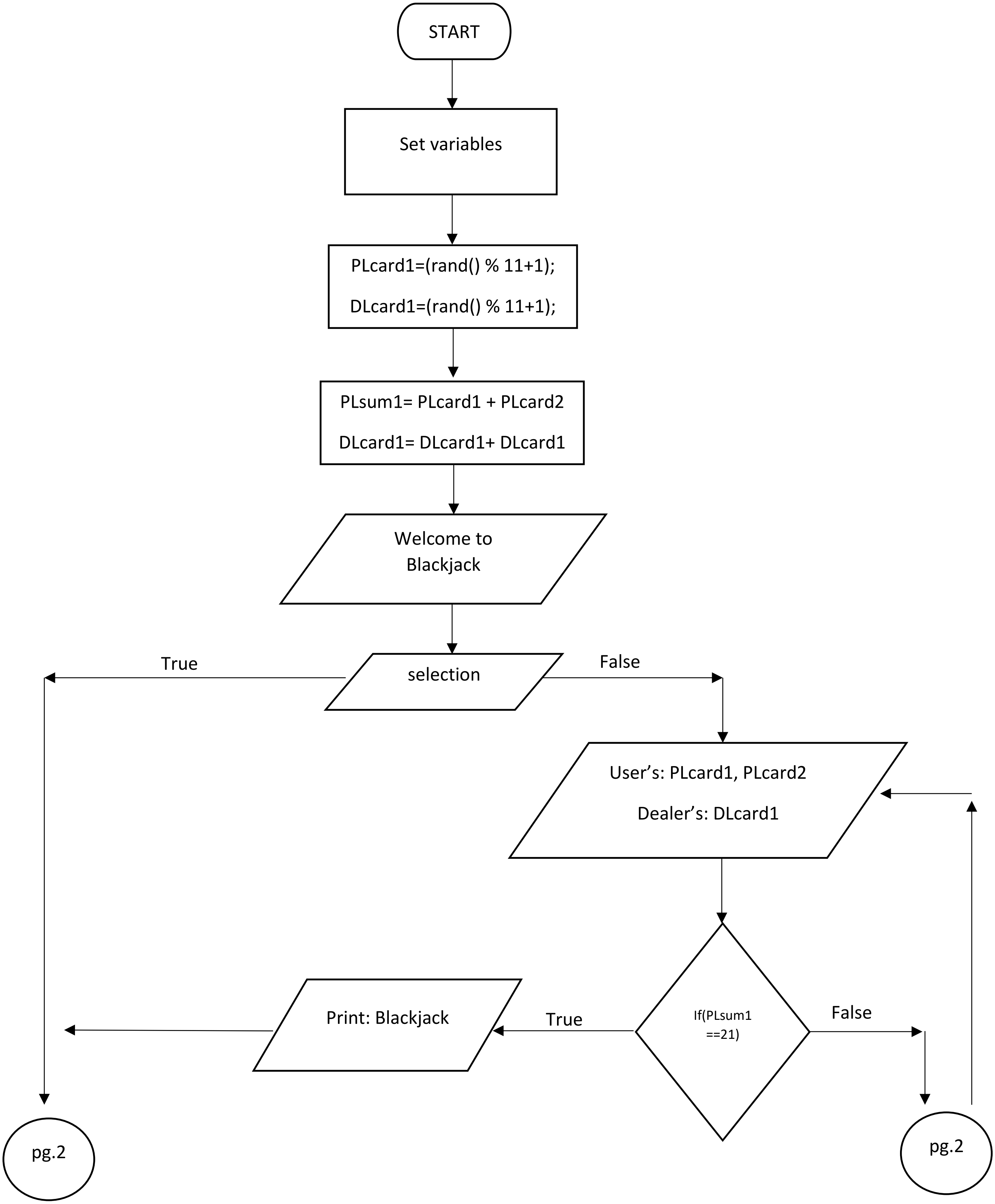


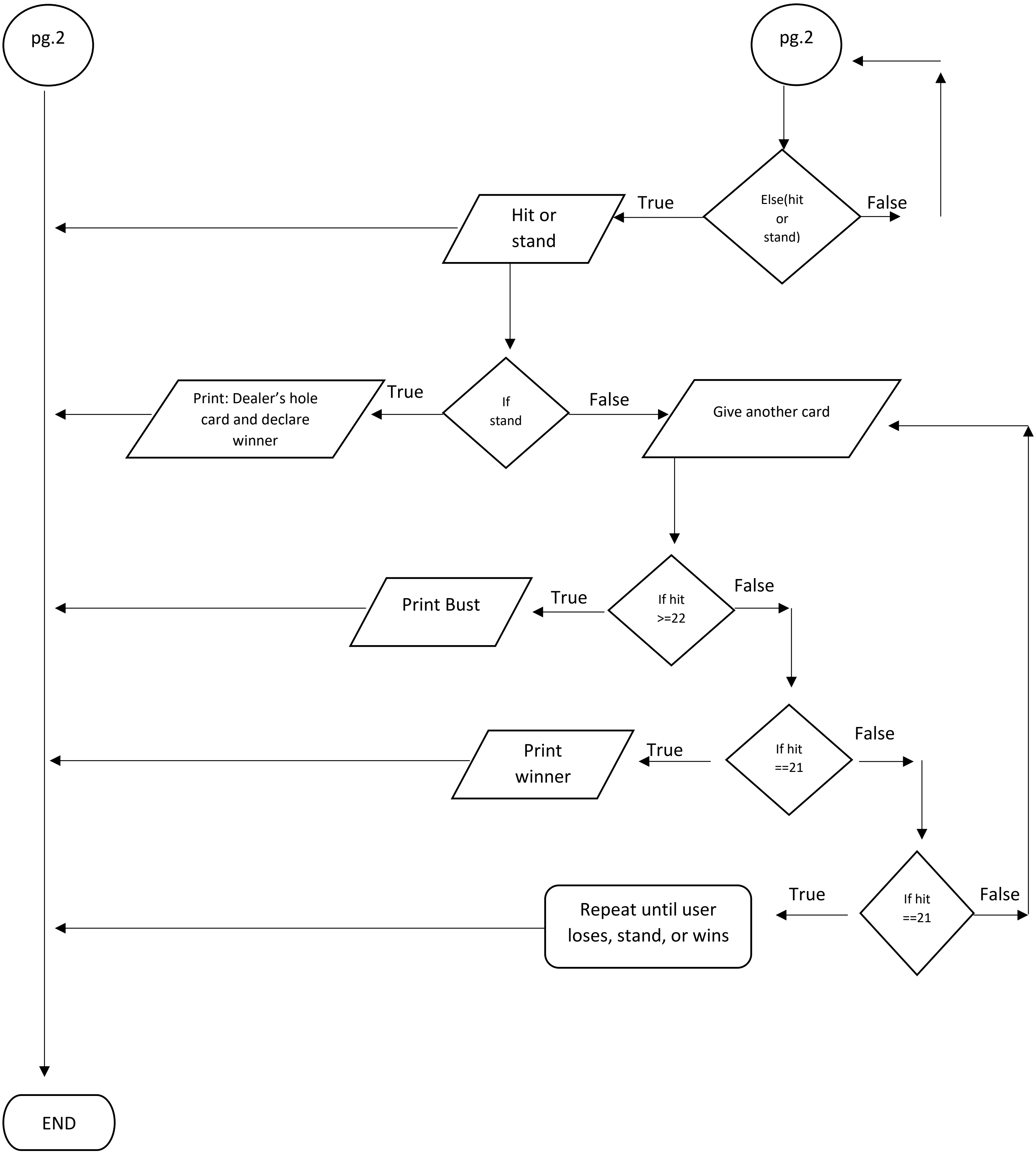
```
BLACKJACK
Welcome to Blackjack!
1. DEAL
2. EXIT
1
Your Cards:
6, 2
                        Your Hand: 8
Dealer's First Card: 6
                        Dealer's Hand: 6
Would you like to hit or stand?
1. HIT
2. STAND
```

PseudoCode

```
1. Enter variables to be inputted by user
2. Enter player’s random generated numbers to equal the card value
    a. Ex. PLcard1=(rand()%11+1)
3. Enter dealer’s random generated numbers to equal the card value
    a. Ex. DLcard1=(rand()%11+1)
4. Enter player’s sum of random generated numbers
    a. Ex. PLsum1 = PLcard1 + PLcard2
5. Enter dealer’s sum of random generated numbers
    a. Ex. DLsum1 = DLcard1 + DLcard2
6. Print title of the game
7. Ask user to deal or exit
8. If deal, start the game
    a. Output user’s two cards
    b. Output dealer’s first card
    c. If user’s get s blackjack
        i. Print: “BLACKJACK”
    d. Ask user to hit or stand
        i. Stand:
            1. Print: Dealer’s hole card and declare winner
        ii. Hit: Give use another card
            1. If user’s card value is >= 22
                a. Print: “BUST”
            2. If user’s card value is == 21
                a. Print: “Winner”
            3. If user’s card value is <21
                    i. Repeat 8d. until user loses, stands, or wins
9. If exit, terminate the game
```

Flowchart


START
Set variables
PLcard1=(rand() % 11+1);
DLcard1=(rand() % 11+1);
PLsum1= PLcard1 + PLcard2
DLcard1= DLcard1+ DLcard1
Welcome to
Blackjack
selection
True
False
User's: PLcard1, PLcard2
Dealer's: DLcard1
If(PLsum1
==21)
True
False
Print: Blackjack
pg.2
pg.2


pg.2
pg.2
Else(hit or stand)
True
False
Hit or stand
If stand
True
False
Print: Dealer's hole card and declare winner
Give another card
If hit >=22
True
False
Print Bust
If hit ==21
True
False
Print winner
If hit ==21
True
False
Repeat until user loses, stand, or wins
END

Checkoff List:

Chapter	Section	Topic	Where Line #"s	Pts	Notes
2	2	cout			
	3	libraries	8,9,10,11,12,13,14	8	iostream, iomanip, cmath, cstdlib, fstream, string, ctime
	4	variables/literals			No variables in global area, failed project!
	5	Identifiers			
	6	Integers	29,30	3	
	7	Characters		3	
	8	Strings		3	
	9	Floats No Doubles	31-34	3	Using doubles will fail the project, floats OK!
	10	Bools	35	4	
	11	Sizeof *****			
	12	Variables 7 characters or less			All variables <= 7 characters
	13	Scope ***** No Global Variables			
	14	Arithmetic operators			
	15	Comments 20%+		5	Model as pseudo code
	16	Named Constants			All Local, only Conversions/Physics/Math in Global area
	17	Programming Style ***** Emulate			Emulate style in book/in class repositiory
3	1	cin			
	2	Math Expression			
	3	Mixing data types ****			
	4	Overflow/Underflow ****			
	5	Type Casting		4	
	6	Multiple assignment *****			

	7	**Formatting output**		**4**	
	8	**Strings**		**3**	
	9	**Math Library**		**4**	**All libraries included have to be used**
	10	**Hand tracing ********			
4	**1**	**Relational Operators**			
	2	**if**		**4**	**Independent if**
	4	**If-else**	**73**	**4**	
	5	**Nesting**		**4**	
	6	**If-else-if**	**100**	**4**	
	7	**Flags *******			
	8	**Logical operators**		**4**	
	11	**Validating user input**		**4**	
	13	**Conditional Operator**		**4**	
	14	**Switch**		**4**	
5	**1**	**Increment/Decrement**		**4**	
	2	**While**	**64**	**4**	
	5	**Do-while**	**65**	**4**	
	6	**For loop**		**4**	
	11	**Files input/output both**		**8**	
	12	**No breaks in loops ********			**Failed Project if included**
******** Not required to show**			**Total**	**100**	